Wright State University

CORE Scholar

[Browse all Theses and Dissertations](#)

[Theses and Dissertations](#)

2016

Probabilistic SVM for Open Set Automatic Target Recognition on High Range Resolution Radar Data

Jason Daniel Roos
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all

Part of the [Electrical and Computer Engineering Commons](#)

Repository Citation

Roos, Jason Daniel, "Probabilistic SVM for Open Set Automatic Target Recognition on High Range Resolution Radar Data" (2016). *Browse all Theses and Dissertations*. 1566.
https://corescholar.libraries.wright.edu/etd_all/1566

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.

Probabilistic SVM for Open Set Automatic Target Recognition on High Range Resolution Radar Data

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

by

Jason D Roos
B.S.E.E., Wright State University, 2014

2016
Wright State University

Wright State University
GRADUATE SCHOOL

August 26, 2016

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Jason D Roos ENTITLED Probabilistic SVM for Open Set Automatic Target Recognition on High Range Resolution Radar Data BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Electrical Engineering.

Arnab Shaw
Thesis Director

Brian Rigling
Chair, Department of Electrical Engineering

Committee on
Final Examination

Arnab Shaw Ph.D.

Brian Rigling Ph.D.

Michael Saville Ph.D.

Robert E.W. Fyffe, Ph.D.
Vice President for Research and
Dean, School of Graduate Studies

ABSTRACT

Roos, Jason D. M.S.E.E., Department of Electrical Engineering, Wright State University, 2016.
Probabilistic SVM for Open Set Automatic Target Recognition on High Range Resolution Radar Data.

The application of Automatic Target Recognition (ATR) on High Range Resolution (HRR) radar data in a scenario that contains unknown targets is of great interest for military and civilian applications. HRR radar data provides greater resolution of a target as well as the ability to perform ATR on a moving target, which gives it an advantage over other imaging systems. With the added resolution of HRR comes the disadvantage that a change in the aspect angle or orientation results in greater changes in the collected data, making classical ATR more difficult. Closed set ATR on HRR radar data is defined when all potential targets are assumed to be part of the training target data base. Closed set ATR has been able to achieve higher rates of correct classification by the selection of proper feature extraction algorithms, however, only a few methods for performing open set ATR have been developed. Open set ATR is the ability to identify and discard when a target is not one of the trained targets. By identifying these untrained targets, the number of misclassified targets is reduced, thereby, increasing the probability of a correct classification in a realistic setting. While the open set ATR produces a more realistic approach, the classical closed-set ATR is the standard method of ATR. One of the more popular classification algorithms currently used today is the Support Vector Machine (SVM). The SVM by nature only works on a binary closed-set problem. However, by extracting probabilities from an SVM as proposed by Platt [1], this classification algorithm can be applied to open set.

In this thesis, the feature extraction methods established in closed-set ATR are modified to facilitate the application of the Probabilistic Open Set Support Vector Machine (POS-SVM). Utilizing the Eigen Template (ET) and Mean Template (MT) feature extraction methods developed for closed-set ATR, in combination with centroid alignment, an open set ATR Probability of correct classification (PCC) rate of 80% has been achieved.

Utilizing POS-SVM, it is possible to successfully perform open set ATR on HRR data with a high PCC .

List of Symbols

Chapter 2

c	Speed of Light
B	Bandwidth
δR	Range Resolution
γ	Chirp Rate
ω	Angular Frequency
I	In Phase Amplitude
Q	Quadrature Phase Amplitude
r	Range Bin
a	Profile Angle
x	HRR Chip
K	Number of Range Bins within HRR Chip
H	Number of Profile Angles within HRR Chip
Σ	Eigen Values
U	Range Subspace
V^*	Angle Subspace
$*$	Conjugate Transpose
$EV[]$	Eigen Vectors
R	Correlation Matrix
y	Class Labels corresponding to feature vectors
H_0	Optimal Hyperplane
w	Hyperplane Normal Vector
b	Offset of Hyperplane from Origin
H_-	Hyperplane closest to data set with class label -1
H_+	Hyperplane closest to data set with class label +1
d	Distance between H_- and H_0 or H_+ and H_0
M	Margin or Distance between H_- and H_+
$\mathcal{L}$	Lagrange Multiplier
L	Number of Feature Vectors to be tested
ζ	error
C	Constant relating error and Margin Size
ϕ	Mapping Function
$K([], [])$	Kernel Function
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
A	Accuracy
$\mathcal{R}$	Recall
F_1	F_1 Score
S	Specificity
Y	Youdens Index

Chapter 4

N	Number of Divisions to apply to HRR Chip
M	Number of Features to be used During Training
K	Size of Training Window in Degrees

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Research Goals	3
1.3	Outline	3
2	Background	4
2.1	HRR	4
2.2	Feature Extraction Methods	7
2.2.1	Mean Template	8
2.2.2	Eigen Template	10
2.3	Support Vector Machines	12
2.3.1	Linear SVM	12
2.3.2	Non Seperable Linear SVM	15
2.3.3	Non Linear SVM	18
2.3.4	Multi Class SVM	20
2.4	Open Set Recognition	22
2.4.1	Probabilistic SVM	23
2.5	Performance Metrics	26
2.5.1	Closed Set Metrics	27
2.5.2	Open Set Metrics	29
3	Methodology	31
3.1	Data Preparation	31
3.2	Centroid Alignment	36
3.3	Training	38
3.4	Validation	42
3.5	Testing	42
3.6	Kernel Selection	44
3.7	Threshold Selection	44

4	Algorithm Evaluation	47
4.1	Closed Set	47
4.2	Open Set	66
5	Conclusion	93
	Bibliography	94

List of Figures

2.1	Example HRR Profile	5
2.2	HRR Profile Collection	7
2.3	Linear SVM Example	13
2.4	Non Separable Linear SVM Example	17
2.5	One vs One Multi-class Classifier (OvO)	21
2.6	One vs All Multi-class Classifier (OvA)	22
2.7	SVM Score Distribution [1]	24
2.8	SVM Score CDF fit to Sigmoid Function[1]	25
2.9	Reduced Open Set Confusion Matrix	30
3.1	Non Uniform Angle Distribution	32
3.2	MSTAR HRR Chip	34
3.3	Sliding Window Implementation	35
3.4	Chip Division	36
3.5	Centroid Alignment	37
3.6	Training Diagram	40
3.7	Training Class Creation in OvA configuration	41
3.8	Testing Diagram	43
3.9	SVM Kernel Optimization	46
4.1	Chip Division and Number of Training Features vs Accuracy for SVM - EigenTemplate	49
4.2	Chip Division and Number of Training Features vs Accuracy for SVM - Mean Template	50
4.3	Chip Division and Number of Training Features Accuracy vs Number of Utilized Angles	51
4.4	Accuracy vs Number of Features for SVM - Eigen Template	52
4.5	Accuracy vs Number of Features for SVM - Mean Template	53
4.6	Accuracy vs Window Size for the SVM-Eigen Template	54
4.7	Accuracy vs Window Size for the SVM - Mean Template	55
4.8	Accuracy vs Number of Test Profiles SVM-Eigen Template	57
4.9	Accuracy vs Number of Test Profiles SVM - Mean Template	58
4.10	Closed set Accuracy and F1 Score	61

4.11	SVM - Eigen Template Matched Filter Confusion Matrix [2]	63
4.12	Chip Division and Number of Training Features vs Known Target Accuracy for POS-SVM Eigen Template	67
4.13	Chip Division and Number of Training Features vs Unknown Target Accuracy for POS-SVM Eigen Template	68
4.14	Chip Division and Number of Training Features vs Known Target Accuracy for POS-SVM Mean Template	69
4.15	Chip Division and Number of Training Features vs Unknown Target Accuracy for POS-SVM Mean Template	70
4.16	Accuracy vs Number of Features POS-SVM Eigen Template	71
4.17	Unknown Target Accuracy vs Number of Features when trained with POS-SVM Eigen Template Data	72
4.18	Accuracy vs Number of Features POS-SVM Mean Template	73
4.19	Unknown Target Accuracy vs Number of Features when trained with Mean Template Data	74
4.20	Accuracy vs Window Size POS-SVM Eigen Template	75
4.21	Accuracy vs Window Size for the POS-SVM Mean Template	76
4.22	Accuracy vs Number of Test Profiles POS-SVM - Eigen Template	77
4.23	Accuracy vs Number of Test Profiles POS-SVM - Mean Template	78
4.24	Known and Unknown Accuracy vs Threshold for POS-SVM - Eigen Template	80
4.25	Known and Unknown Accuracy vs Threshold for POS-SVM - Mean Template	81
4.26	ROC Curve for POS-SVM - Eigen Template by Target	82
4.27	ROC Curve for POS-SVM - Mean Template by Target	83
4.28	Averaged ROC Curve for POS-SVM ET and MT	84
4.29	Open Set Accuracy and F1 Score for Known Targets	89
4.30	Open Set Classification Accuracy	90

List of Tables

4.1	Closed Set Tuning Parameters SVM-Eigen Template	59
4.2	Closed Set Tuning Parameters SVM - Mean Template	59
4.3	Closed Set Confusion Matrix for SVM-Eigen Template Features	60
4.4	Closed Set Confusion Matrix for SVM - Mean Template Features	60
4.5	Closed Set Accuracy and F1 Score	62
4.6	Closed Set SVM - Eigen Template Confusion Matrix for 5 Targets	64
4.7	Closed Set SVM - Eigen Template Confusion Matrix for 6 Targets	64
4.8	Closed Set SVM - Eigen Template Confusion Matrix for 7 Targets	65
4.9	Open Set POS-SVM - Eigen Template Tuning Parameters	85
4.10	Open Set POS-SVM - Mean Template Tuning Parameters	85
4.11	Open Set Confusion Matrix POS-SVM - Eigen Template	86
4.12	Open Set Confusion Matrix POS-SVM - Mean Template	86
4.13	Reduced Open Set Confusion Matrix POS-SVM - Eigen Template	87
4.14	Reduced Open Set Confusion Matrix POS-SVM - Mean Template	87
4.15	Open Set Statistics	90
4.16	Open Set Confusion Matrix 4 Known and 1 Unknown POS-SVM - Eigen Template	91
4.17	Open Set Confusion Matrix 5 Known and 1 Unknown POS-SVM - Eigen Template	91
4.18	Open Set Confusion Matrix 4 Known and 2 Unknown POS-SVM - Eigen Template	92
4.19	Open Set Confusion Matrix 3 Known and 3 Unknown POS-SVM - Eigen Template	92

Acknowledgment

I would like to take this opportunity to extend my thanks to my friends and family in their support and patience as I performed my research. Also I would like to extend my thanks to my committee members for their valuable feedback and willingness to serve on my committee. Additionally I would like to extend a special thanks to Dr. Shaw for his guidance during the past three semesters and the hours spent in his office reviewing the research, as well as for his help in writing and editing the document. Finally and most of all I would like to thank my wife who has constantly encouraged me and has been supportive patient and loving throughout the entire process.

Dedicated to
My Wife Jenny

1. Introduction

1.1 Motivation

Automatic target recognition (ATR) is a topic of great interest for military and other government and civilian applications. The ability for a computer to identify a target from data collected by a sensor is valuable in terms of time and money. The applications for ATR continue to grow as new sensors and methods of collecting data increases. For instance, areas in security such as facial recognition and license plate scanning have benefited from advances in ATR. Although ATR has progressed substantially with the digital revolution, the ability to automatically detect a target with a computer instead of a well trained human analyst is not a trivial task. Due to the similarities in appearances between various targets, substantial research has been focused on improving sensors to be able to differentiate between targets as well as developing algorithms to automatically recognize these targets.

One sensor that has been of particular interest in military applications is High Range Resolution (HRR) radar. HRR is a modification to radar in order to compensate for motion blurring due to the movement of a target. As indicated by its name, HRR has a high range resolution. By utilizing HRR data, information can be collected with the sensor that would allow the distinction between targets of similar size, shape, and purpose, even if that target is moving. One trade off with this method, however, is that HRR signatures can vary based on the viewing aspect of the target. This makes HRR ATR more challenging when compared to ATR on an image of the full target such as SAR.

Performing ATR on HRR data has been a topic of research for more than a decade with various feature extraction methods being proposed and tested [3][2][4][5][6][7][8][9][10][11]. The ability to correctly identify a target from a trained library of targets has been well researched. Many of these algorithms are able to correctly identify a target with a high degree of accuracy under ideal test conditions. Much of the recent work in HRR ATR has centered around utilizing support vector machines (SVMs) as a means of classifying targets [5] [6][12][13][14]. Currently SVM's have performed well, with the added benefit of computationally efficient classification after training.

Recently, an improvement to classical or closed-set ATR has been developed. This new ATR method, referred to as open set, allows for the declaration of a target as being unknown. This opens up to the possibility of identifying previously unknown targets for further investigation. The ability to classify a target as unknown has been gaining popularity due to the large benefit in military and civilian applications. Research has been done with applying open set theory to HRR [15]; however, the application of open set theory with an SVM classifier, which has shown good results with other data types [16][17][18][19][20], has not been performed.

Utilizing the work performed by Scherreik [16] on open set recognition as well as the feature extraction algorithms developed by Shaw [2] and Williams [3], a new method of applying open set ATR on HRR data is developed. Results based on the MSTAR data set show improvement in the identification of unknown targets consistent with similar work performed on SAR data [16]. In addition, the closed-set ATR performance for the two feature extraction methods demonstrate improvement when utilizing the SVM over each of the method's respective classification algorithms.

1.2 Research Goals

Several methods have been developed to extract features from HRR signatures, even though few have been applied to open set recognition (OSR). This research will look into applying feature extraction algorithms for closed-set ATR on HRR data and implementing them with the Probabilistic Open Set Support Vector Machine (POS-SVM), which has been identified as a successful open set ATR classification algorithm [16]. This algorithm will be validated using the MSTAR HRR data set with various configurations of the developed algorithm. In addition, performance metrics and results will be provided for validation and comparison of methods.

1.3 Outline

This thesis is divided into five chapters. Chapter 2 is a review of the various elements studied in the thesis and an investigation into their derivations. Chapter 3 provides an in-depth explanation of the methods that were used to implement the proposed ATR algorithm. Finally, an in-depth look at how the algorithm performed when compared to previous algorithms is given in Chapter 4.

2. Background

2.1 High Range Resolution Radar

The amount and quality of information that a radar can gather from the environment is a function of the radar's resolution among other features of the radar. Due to the constant desire for better resolution, new ways of utilizing radar have been studied. Early radar resolution allowed for the general direction and range of incoming objects; however, the ability to differentiate and identify the objects was limited due to the technology of the day. Upon discovery of the effectiveness of radar as an early warning platform during World War II (WWII), research was devoted to improving the abilities of the radar along with discovering new applications. After the end of WWII, a new method for utilizing radar was discovered that differed from the typical radar usage of ground based monitoring. In the 1950's, Synthetic Aperture Radar (SAR) was developed [21] and brought about the ability to form 2 dimensional (2-D) images of the terrain when used in an aircraft. SAR collects returns in range and cross range similar to that of an image taken with a camera, where the cross range would be the x-axis of an image and the range the y-axis. The main advantage to SAR over other imaging techniques was its ability to collect the image invariant of light and weather. One disadvantage to SAR as well as many other imaging techniques is the inability to resolve an object that is moving while the image is being collected.

HRR was developed in order to compensate for the blurring caused by a moving target in SAR. The way in which HRR collects information about a target is different. Instead of

range and cross-range, HRR collects data in range and angle. This means that the information will appear as a 1 Dimensional (1-D) return. The strength and location in range of the return will be dependent on how the reflections from the target surface combine coming back along the specified line of sight angle of the radar over the pulse duration [22]. While the 1-D return will compensate for issues such as blurring, it also brings about issues of its own. In the dissertation by Nelson, he states that HRR profiles “add constructively or destructively depending on their relative phases. Therefore a slight change in the relative phases in the returns can have significant effect on the signature.” [22]. This means that the aspect of the target can greatly impact the actual HRR signature. An example, 1-D HRR profile can be seen in figure 2.1.

Figure 2.1: Example HRR Profile

The range resolution of HRR is dependent on the bandwidth of the radar system [23], where the higher the bandwidth the better the resolution. The relationship between bandwidth and range resolution is given in equation 2.1. In this equation, c is the speed of light in a vacuum and B is the bandwidth of the sensor.

$$\delta R = \frac{c}{2B} \quad (2.1)$$

While having a good range resolution is important for HRR, another important aspect of HRR radar is its ability to resolve a change in frequency such as a Doppler shift and the maximum distance the radar can resolve a target [23]. The chirp signal seen in equation 2.2 can be tuned to have a greater bandwidth or range resolution by increasing the chirp rate or γ . This in turn increases the amount of bandwidth the signal occupies without forfeiting detection range or Doppler sensitivity, which rely on the ability to distinguish the return time of the wave as well as the change in frequency experienced by the wave.

$$s(t) = \begin{cases} e^{j(\omega_0 t + \gamma t^2)}, & \text{if } |t| \leq T_0/2 \\ 0, & \text{if } |t| \geq T_0/2 \end{cases} \quad (2.2)$$

As technology continues to improve and new radar waveforms are developed that allow for better range and Doppler resolution, the ability for a HRR radar to resolve more information from a target thereby increasing its usefulness as an ATR sensor will continue to improve as well.

It is important to understand some terminology used when talking about HRR data. A profile in HRR is a 1-D vector consisting of complex data, where every profile is collected at some angle dictated by the offset from the normal of the collection antenna array. A collection of HRR profiles over a set of profile angles is a chip. A chip is collected at an angle relative to the target. For example, a chip angle of 0° could refer to the front of a target with a chip angle of 180° referring to the rear of the target. An example of these

terms can be seen in figure 2.2.

Figure 2.2: HRR Profile Collection

2.2 Feature Extraction Methods

The goal of the study of ATR as well as any other machine learning application is relatively simple. Extract information from a given set of data such that the information is unique when compared to information gathered in the same way from other unique sets of data. In ATR, information is often referred to as features and the various sets of data are referred to

as classes. Classes generally represent physical objects, such as a car, boat, or tank, which need to be differentiated programatically in an automated manner. Additionally in ATR, the process used to create features from given classes of data is referred to as feature extraction since class features are generally calculated or extracted from a representative collection of classes. The selection of the feature extraction algorithm is generally driven by the method utilized to collect the information about a class, generally meaning that a different feature extraction method is used for different collection methods. Below are examples of two different feature extraction algorithms that have been shown in literature to work well with HRR data.

2.2.1 Mean Template

As given in [24], one method for extracting features from HRR data is to average a chip of HRR profiles. This method is referred to as the Mean Template (MT) method. When HRR data is collected, it has I and Q data representing the in-phase amplitude (I) and the quadrature-phase amplitude (Q). The MT method computes the average of the magnitude of the return of the sensor, where the magnitude of the data is calculated using equation 2.3.

$$x = \sqrt{I^2 + Q^2} \quad (2.3)$$

When comparing HRR data, it is important that all of the data is normalized by the mean signature power to “remove automatic gain control and range effects.” [24]. The normalization of the HRR profiles is performed utilizing equation 2.4, where a is the profile angle, r is a range bin, x is an HRR Chip, K is the number of range bins, and H is the number of profile angles per chip.

$$x = \begin{bmatrix} r_{11} & r_{12} & r_{13} & \dots & r_{1H} \\ r_{21} & r_{22} & r_{23} & \dots & r_{2H} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{K1} & r_{K2} & r_{K3} & \dots & r_{KH} \end{bmatrix}$$

$$\vec{a}_1 = [r_{11}, r_{21}, \dots, r_{K1}]^T$$

$$\bar{r}_{ij} = \frac{r_{ij}}{\frac{1}{K}|\vec{a}_j|^2} \quad i = 1, 2, \dots, K; \quad j = 1, 2, \dots, H \quad (2.4)$$

$$\bar{x} = \begin{bmatrix} \bar{r}_{11} & \bar{r}_{12} & \bar{r}_{13} & \dots & \bar{r}_{1H} \\ \bar{r}_{21} & \bar{r}_{22} & \bar{r}_{23} & \dots & \bar{r}_{2H} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{r}_{K1} & \bar{r}_{K2} & \bar{r}_{K3} & \dots & \bar{r}_{KH} \end{bmatrix}$$

In addition to normalizing the data, it is desired that the data follow a normal distribution to improve the performance of the classifier [24]. A power coefficient of .14 is applied to every range bin in the chip. This transforms each HRR profile to have a normal distribution, where the power transform is given in equation 2.5.

$$X = \begin{bmatrix} \bar{r}_{11}^{(.14)} & \bar{r}_{12}^{.14} & \bar{r}_{13}^{.14} & \dots & \bar{r}_{1H}^{.14} \\ \bar{r}_{21}^{.14} & \bar{r}_{22}^{.14} & \bar{r}_{23}^{.14} & \dots & \bar{r}_{2H}^{.14} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{r}_{K1}^{.14} & \bar{r}_{K2}^{.14} & \bar{r}_{K3}^{.14} & \dots & \bar{r}_{KH}^{.14} \end{bmatrix} \quad (2.5)$$

Once the collection of HRR profiles has been normalized, the angle values are averaged such that only a single vector is produced for each collection of HRR profiles as is

seen in equation 2.6.

$$\vec{X}_i = \frac{1}{H} \cdot \sum_{j=1}^H X_{i,j} \quad i = 1, 2, \dots, K \quad (2.6)$$

This is repeated for every chip of HRR profiles in both the training, validation, and testing data sets.

2.2.2 Eigen Template

Another and more promising feature extraction method for HRR data is the Eigen Template (ET) method [2]. This method also utilizes equation 2.3 to extract the magnitude values from a HRR chip. Once the magnitude of the HRR profiles has been returned, the ET method power transforms the chip as seen in equation 2.7, where a is the profile angle, r is a range bin, x is an HRR Chip, K is the number of range bins, and H is the number of profile angles per chip..

$$\bar{x} = \begin{bmatrix} \bar{r}_{11} & \bar{r}_{12} & \bar{r}_{13} & \dots & \bar{r}_{1H} \\ \bar{r}_{21} & \bar{r}_{22} & \bar{r}_{23} & \dots & \bar{r}_{2H} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \bar{r}_{K1} & \bar{r}_{K2} & \bar{r}_{K3} & \dots & \bar{r}_{KH} \end{bmatrix} \quad \bar{r}_{ij} = r_{ij}^{(.14)} \quad (2.7)$$

$$\vec{a}_1 = [\bar{r}_{11}, \bar{r}_{21}, \dots, \bar{r}_{K1}]^T$$

Once the HRR chip has been power transformed, it now has a normal distribution; however, the chip must still be normalized. The normalization performed in the ET method is different than that of the MT method. Instead of normalizing by the mean signature power, each HRR profile within a chip is normalized by its respective L_2 norm as seen in equation 2.8. This allows for the direct comparison of the ET feature vectors by the classifier.

$$r'_{ij} = \frac{\bar{r}_{i,j}}{|\vec{a}_j|} \quad i = 1, 2, \dots, K; \quad j = 1, 2, \dots, H \quad (2.8)$$

$$X = \begin{bmatrix} r'_{11} & r'_{12} & r'_{13} & \dots & r'_{1H} \\ r'_{21} & r'_{22} & r'_{23} & \dots & r'_{2H} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r'_{K1} & r'_{K2} & r'_{K3} & \dots & r'_{KH} \end{bmatrix}$$

The eigen vector acts as an estimation of the correlation matrix of the HRR training profile. Shaw in [2] showed that the correlation matrix of a collection of HRR profiles can be described by equation 2.9, where X is the normalized and power transformed collection of HRR profiles. To estimate the eigen vectors of the correlation matrix, Shaw went on to show that with the SVD decomposition of X can be decomposed into its eigen values (Σ) as seen in equation 2.10. It can also be decomposed into the eigen vectors for the range subspace (U) and angle subspace (V^*), where $*$ is a conjugate transpose. In equation 2.11, it can be seen that U is equivalent to the eigen vectors of the correlation matrix, where $EV[]$ is the eigen vectors. HRR profiles are independent in V^* and so U is used to create the feature vector.

$$\hat{R} = XX^T \quad (2.9)$$

$$X = U\Sigma V^* = \sum_{i=1}^M u_i \lambda_i v_i^T \quad (2.10)$$

$$U = EV[XX^T] \quad (2.11)$$

Based upon the work done in [2][7][9][10][11], it is also seen that the majority of the information of the eigen vectors are contained within the first eigen vector of U [2]. As a result of this, only the first eigen vector is used to create the ET from the inputted collection of HRR profiles. The ET method reduces a single collection of HRR profiles from $K \times H$

to $K \times 1$, where K is the number of range bins and H is the number of profile angles.

2.3 Support Vector Machines

The majority of machine learning algorithms require that the data being classified has some amount of separation between classes in the feature space. Due to the nature of HRR data, the number of features or dimensions for a single collection of data is quite large. As a result of the dimensionality of this data, it is important that a classification algorithm is chosen that does not degrade with an increase in dimensionality. Currently, the support vector machine (SVM)[25] has been shown to be a very successful classification algorithm on a wide range of data sets. In addition to popularity, the SVM is also “immune to dimensionality” [26], which is a common problem with HRR data. This makes the SVM a good choice for HRR ATR.

2.3.1 Linear Support Vector Machine

The classical or linear SVM is constructed by performing an optimization of a N dimensional plane, or hyper plane, that equally separates two classes of data in the feature space. The first step in an optimization problem is to make an initial guess of the solution. In order to make the initial guess of the hyper plan the inputted data or training data x is grouped by class y such that for every feature vector x_i there is a class label y_i that identifies to which class x_i belongs. By nature, the SVM is a binary classifier which requires that y contains only two elements as given in 2.12.

$$y \in \{-1, 1\} \tag{2.12}$$

Figure 2.3: Linear SVM Example

Once the data has been separated, the next step is to find a hyperplane H_0 that equally divides the two classes in the feature space, as in figure 2.3, where the hyperplane is described by equation 2.13. In this equation, w is a normal vector to the hyperplane and b is the offset of the hyperplane from the origin.

$$H_0 = w \cdot x + b = 0 \quad (2.13)$$

The overall goal of the SVM is to maximize the distance from H_0 to the closest points in each class. In order to find the closest points from each class to the hyper plane, specify two more planes parallel to H_0 , where one plane is defined by the closest point x_i such that $y_i = 1$. This plane is H_+ and for $y_i = -1$ the corresponding plane is H_- . Since H_+ and H_- are determined by the point(s) closest to H_0 , the objective is to maximize the distance from H_+ to H_0 . This distance can be found using equation 2.14, where w is the vector normal to H_0 and therefore H_+ and H_- . The x_i associated with the smallest d are

generally referred to as the support vectors of the hyperplane.

$$d = \min_{i=1 \dots N} \frac{||w \cdot x_i + b||}{||w||} \quad (2.14)$$

Going back to the definition of H_0 , d is the same for both H_+ and H_- , meaning that the distance between H_+ and H_- can be found using equation 2.15 and is referred to as the margin.

$$M = \frac{2}{||w||} \quad (2.15)$$

When maximizing M , it is important that the division of the classes defined by y is maintained. This is done by creating conditions given in equations 2.16 and 2.17 for $y_i = 1$ and $y_i = -1$ respectively.

$$w \cdot X_i + b \geq 1 \quad (2.16)$$

$$w \cdot X_i + b \leq -1 \quad (2.17)$$

Since the maximization of equation 2.15 is not straight forward, it is simpler to rewrite it as a minimization problem. The minimization is seen in equation 2.18, where the condition is given in 2.19.

$$\frac{d}{dw} M = 0 \equiv \min \left\{ \frac{||w||^2}{2} \right\} \quad (2.18)$$

$$y_i(w \cdot x_i + b) - 1 \geq 0 \quad (2.19)$$

While the solution of equation 2.18 is still not straight forward, if the method of Lagrange multiplier is applied to the problem, a solution can be found. The Lagrange method allows for the optimization of the margin to be rewritten as 2.20, where λ is referred to as the Lagrange multiplier.

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2} ||w||^2 - \sum_{i=1}^N \lambda_i (y_i(w \cdot x_i + b) - 1) \quad (2.20)$$

Once the problem is converted to the Lagrangian method the next step is to take the partial derivative of $\mathcal{L}$ with respect to w and b and set to zero. The resulting equations can be seen in 2.21 and 2.22 referenced from [26].

$$w = \sum_{i=1}^N \lambda_i y_i x_i \quad (2.21)$$

$$b : \sum_{i=1}^N \lambda_i y_i = 0 \quad (2.22)$$

Plugging 2.21 and 2.22 into equation 2.20, the optimization now takes the form of 2.23 with the constraints that $\lambda_i > 0$ and $\sum \lambda_i * y_i = 0$.

$$\mathcal{L}(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j x_i^T x_j \quad (2.23)$$

The optimized w and b can be found by maximizing 2.23, which minimizes 2.20. This maximizes the margin of the SVM, which was the original intent.

Once the SVM has been trained, any new test point $\bar{x}_l$ can be evaluated to determine to which class it belongs to with equation 2.24, where *sign* returns a -1 if the result is negative and a $+1$ if the result is positive, where L is the number of vectors to be tested.

$$f(x) = \text{sign}\left(\sum_{i=1}^L \lambda_i y_i x_i \cdot \bar{x}_j^T + b\right) \quad (2.24)$$

This function is often referred to as the decision function 2.24 and is computationally efficient making the SVM an efficient classification algorithm.

2.3.2 Non Seperable Linear Support Vector Machine

The case of the linear SVM assumes an ideal environment, where the two classes of data can be perfectly separated with a plane. However, this is often not the case in practice.

This results in a non ideal margin as well as an intensive computational process. To handle the case when the data is not perfectly separable, a modification to the linear SVM can be applied [26]. In equation 2.19, a condition for the minimization of the SVM is given. If an error denoted by ζ is included in this condition, then the outliers in the two classes that do not represent the general location of the classes can be ignored in order to increase the margin size. This new condition is given in 2.25. Equation 2.20 now becomes 2.26, where C controls the trade off between margin size and the number of ζ . Figure 2.3.2 gives an example of how the choice of ζ can allow for the selection of H_0 when the classes overlap.

$$y_i(w \cdot x_i + b) - 1 + \zeta \geq 0 \quad (2.25)$$

$$\mathcal{L}(w, b, \lambda) = \frac{1}{2}||w||^2 + C \sum_{i=1}^N \zeta_i - \sum_{i=1}^N \lambda_i (y_i(w \cdot x_i + b) - 1) \quad (2.26)$$

$$b = \frac{1}{y_i} - w \cdot x_i \quad (2.29)$$

A method to select C has no direct solution and an exhaustive search is generally performed to find the ideal solution by cross validation of the optimized SVM margin with the training set, or x . This method identifies which C produces the highest Probability of correct classification (P_{cc}). It can be noted that if enough data exists, an additional set can be created to use as cross validation data. This set is generally referred to as the validation set.

2.3.3 Non Linear Support Vector Machine

Another case considered for the SVM occurs when the classes for the given data are not separable with a line or hyperplane. While the non separable SVM does handle minor issues relating to this idea, the SVM can still perform poorly if the distance between the support vectors for each class is small or the value for C is large. To accommodate this kind of data a nonlinear SVM is derived, where the data is fit to a nonlinear function and the ideal margin is then found. However, since a solution for a nonlinear equation is generally more difficult, the data is mapped into a higher dimension described by a nonlinear equation. The mapping of this data then allows for a plane to be found in the higher dimension that properly separates the classes.

Transforming the data from m dimensions to k dimensions requires a mapping function generally represented as $\phi(x)$, where x has m dimensions and $\phi(x)$ produces a vector with k dimensions. This extension of the linear SVM to nonlinear SVM requires a substitution in equations 2.13 through 2.29. To make an SVM nonlinear, all x are replaced with $\phi(x)$. This does, however, present a computational issue due to the increase from m dimensions to k dimensions. To solve this problem, a different method of mapping the data is implemented that is often called the *kernel trick*.

To understand the kernel trick, it is important that its derivation is discussed. Equation 2.23 finds the optimal Lagrange multipliers based upon the training data x and its matching

classes y . In order to find the ideal margin for the x values in the higher dimensional space, the x values in this equation are replaced with the mapping function $\phi(X)$, thereby solving the SVM for the new k dimensions. The new Lagrange equation for the nonlinear SVM is given in equation 2.30. Where the calculation of the ideal w and b follows the same derivation as the classical SVM, simply replace the x data which existed in m space with $\phi(x)$ in k space.

$$\mathcal{L}(\lambda) = \sum_{i=1}^N \lambda_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \lambda_i \lambda_j y_i y_j \phi(x_i)^T \phi(x_j) \quad (2.30)$$

The solution is straight forward for the nonlinear SVM; however, one issue is apparent. When $k \gg m$, a computational slowdown occurs because for every iteration of the SVM and for every i in x the data has to be transformed into the higher dimensions. To solve this issue, note that $\phi(x_i)^T \phi(x_j)$ is the dot product of the feature vectors and always maps the data to the same feature space. To replicate this calculation, the kernel function is defined as 2.31.

$$K(x_i, x_j) \equiv \phi(x_i)^T \phi(x_j) \quad (2.31)$$

In order for K to be a valid substitution for the dot product, it must first satisfy Mercers theorem [27] which states: “A symmetric function $K(x, y)$ can be expressed as an inner product for some ϕ if and only if $K(x, y)$ is positive semidefinite i.e. $\int K(x, y)g(x)g(y)dx dy \geq 0 \quad \forall g$ ”. This means that if the kernel matrix meets the above criteria, the dot product of the two vectors in the higher dimensional space can be computed without having to map the data into the higher dimensional space. This saves computation time and increases the viability of the SVM as an efficient classification algorithm.

There are many known kernel functions that can be employed with the SVM; however, some of the more common kernel tricks are the Polynomial Kernel seen in equation 2.32

as well as the RBF or Gaussian Kernel seen in equation 2.33.

$$K(x_1, x_2) = (\gamma x_1 \cdot x_2 + c)^\delta \quad (2.32)$$

$$K(x_1, x_2) = \exp\left(\frac{\|x_1 - x_2\|^2}{\sigma^2}\right) \quad (2.33)$$

Both the Polynomial and Gaussian Kernels also require some optimization of their respective coefficients. Currently, the most common method to calculate these coefficients is to use an exhaustive search across a range of values. This is similar to the method used to find the optimal C in section 2.3.2. A better method to calculate the various coefficients for the respective kernel functions is being researched; however, as of the date of this paper, no general solution has been found.

2.3.4 Multiple Class Support Vector Machine

Inherently, the SVM can only make a classification between two classes. In the real world, there are generally more than two classes of targets to be classified with ATR. In this case, a couple strategies can be implemented.

The first strategy is to generate K SVM's, where K is the number of comparisons required to compare each class with every other class. Utilizing this metric, a N class problem will generate $N * (N - 1)/2$ SVM's to make a four class problem require 6 SVM's to fully test all 4 classes. It is convention to call this kind of multi class SVM a one vs one (OvO) strategy. As an example, consider a three class problem, where the classes are A , B , and C . The OvO strategy would train 3 SVM's, where the three SVM's compare the classes as follows: A vs B , A vs C , and B vs C (see Figure 2.5).

Due to the fact that multiple predictions are provided per test feature, a decision metric is required for the OvO classification strategy. Since OvO compares each class to every other class, it is assumed that the true class will be chosen the majority of the time. Know-

Figure 2.5: One vs One Multi-class Classifier (OvO)

ing that the most frequently occurring class should be the true class, the decision function is to take the mode of the predicted classes. While this decision metric works most of the time, a problem arises when the mode contains two or more values. In this case, the mode is multi-modal and the class with the greatest distance from the corresponding SVM margin is chosen.

The second most common strategy is to think of the problem as a determination of whether or not a test feature belongs to a class. This method, while similar to OvO, only utilizes N SVM's in a N class problem, which reduces the complexity. This strategy is commonly called the One vs Rest (OvR) or One vs All (OvA). For simplicity, this strategy will be referred to as One vs All. The OvA strategy converts N classes into 2 classes, where the first class is being tested for and the remaining classes form the second class. Given the example classes for OvO above, the SVM trained with the OvA strategy would produce classes as follows: A vs $\{B \& C\}$, B vs $\{A \& C\}$, and C vs $\{A \& B\}$. The decision metric for the OvA strategy relies on the calculated distance from the hyperplane to the test feature. The SVM that returns the greatest distance is selected as the true class for that test

feature as seen in figure 2.6.

Figure 2.6: One vs All Multi-class Classifier (OvA)

2.4 Open Set Recognition

When set theory is applied to ATR or any pattern recognition problem, there are two major methods for defining the type of problem. If the classes in pattern recognition are equivalent to members in a set, then we can define the set in one of two ways. The classical way of defining an ATR problem is when all of the members of that set are known, meaning that all possible targets have been identified and are used in the training of the classification algorithm. This description of the ATR problem is referred to as a closed-set or forced decision [2] problem. The second ATR problem is when not all of the members of the target set are known. In this case, the set is said to be infinite or an open set since the set is not fully defined. Open set problems in ATR add another level of complexity to the decision making process when compared to closed-set problems [2]. Instead of selecting the most probable

candidate, the decision must now determine if the most probable candidate belongs to one of the known classes. Recently, Open Set Recognition (OSR) has been of great interest, where research was originally done by Scheirer [28] and followed up by Scherrek [16] [17][18][19][20].

The definition for the openness of a classification problem [28] is given in section 2.34, where M , N , and P are the number of training, target, and testing classes respectively and the classification problem is closed set when O is zero [16].

$$O = 1 - \sqrt{\frac{2M}{P + N}} \quad (2.34)$$

Equation 2.34 shows that as the number of unknown classes increases so does the openness, where the number of unknown classes is given by 2.35.

$$U = N - P \quad (2.35)$$

2.4.1 Probabilistic SVM

The SVM classification method described in section 2.3 is only applicable to the closed-set classification method. However, Platt [1] has developed a method to obtain the posterior probability that a score, or the distance from a target to the margin M , belongs to the chosen class. Platt built upon earlier work [25] which showed that the posterior probability of an SVM can be obtained by fitting the scores of an SVM to a sum of cosines. The calculated coefficients of the sum of cosines could then be used to estimate the posterior for a calculated score. Platt enhanced this method to improve the computational efficiency. He showed that while the distribution of the scores of the SVM do not follow the Gaussian distribution, see figure 2.7, the cumulative distribution function (CDF) of these scores can be fit to a sigmoid function. This function is seen in figure 2.8 and is defined in equation

2.36, where A and B are the coefficients.

Figure 2.7: SVM Score Distribution [1]

Figure 2.8: SVM Score CDF fit to Sigmoid Function[1]

$$P(y_i = 1|f(x_i)) \approx \frac{1}{1 + \exp(Af_i + B)} \quad (2.36)$$

The parameters A and B are found by calculating the Maximum Likelihood Estimation (MLE) of the scores f_i obtained by classifying a set of data. Where the classes of the data are known, this set of data is often referred to as the validation set and is separate from the test set. With the posterior probability model defined for a trained SVM, an extension of the SVM used as an open set classification algorithm is now possible. Obtaining the probability that a value $\bar{x}_i$ belongs to a class requires that the ideal values for A and B be found by performing the MLE on equation 2.36. Once the distribution has been estimated, the probability of $\bar{x}_i$ belonging to the identified class is obtained by calculating the distance

from the point $\bar{x}_i$ to the margin hyperplane H_0 found in section 2.3. This score can be found by equation 2.37, where w is the normal vector to the hyper plane, b is the offset of the hyper plane from the origin, and $\bar{x}_i$ is the point to be tested.

$$f(x) = \bar{x}_i \cdot w + b \quad (2.37)$$

With the probability of the test point known it is now possible to estimate when a point should be classified as unknown. While a method for finding the ideal threshold has been proposed by Scherreik [16], these methods were unsuccessful on HRR data upon implementation. For this application a fixed threshold found utilizing a validation set has been found to be ideal. Upon training the SVM, the validation set is run utilizing equation 2.24 and then based upon the classification performance of the validation set, a threshold can be selected.

2.5 Performance Metrics

When talking about the performance of an open set algorithm, it is important to identify metrics which can be used to identify the success of that algorithm. Classification problems are commonly scored using four different metrics. The four metrics are: True Positive (TP), True Negative (TN), False Positive (FP), and a False Negative (FN). The four metrics are calculated from a matrix of size $P \times N$, where P is the number of test classes and N is the number of target classes. This matrix is referred to as the error matrix or more commonly the Confusion Matrix (CM). For closed set, the CM is square and thus $P = N$. The CM is formed by grouping the classification results such that the rows of the matrix represent the actual class of the test data and the columns represent the N possible classes. The results of the classification algorithm are collected such that the value in the matrix represents the number of times a result was classified as belonging to the class specified by

the column, when it actually belongs to the class specified in the row. With the information from a CM, the probability of data being properly classified can be calculated as well as the probability of the data being misclassified.

2.5.1 Closed Set Metrics

As given in section 2.4, a closed-set classification problem is when there are the same number of training classes as there are target classes, making O in equation 2.34 equal to 0. Utilizing a CM constructed from a training and testing set, a determination can be made as to how well a classifier can perform [29]. In section 2.5, four different metrics were provided (TP , TN , FP , and FN). The calculation of these four metrics is given in equations 2.38 - 2.41 for the k -th correct class, where E represents the CM.

$$\text{TruePositive} : \quad TP_k = E_{k,k}; \quad (2.38)$$

$$\text{TrueNegative} : \quad TN_k = \sum_{i=1}^N E_{i,i} - TP_k \quad (2.39)$$

$$\text{FalsePositive} : \quad FP_k = \sum_{i=1}^P E_{i,k} - TP_k \quad (2.40)$$

$$\text{FalseNegative} : \quad FN_k = \sum_{i=1}^N E_{k,i} - TP_k \quad (2.41)$$

With the four metrics above, several statistics about the classification algorithm can be computed. Some of those statistics are: Accuracy, Precision, Recall, and F_1 Score.

2.5.1.1 Accuracy

In order to determine how well the classification algorithm correctly identified the test set, the accuracy needs to be calculated. Accuracy is calculated in equation 2.42, where A is the average accuracy of the CM. It indicates how well the test set was classified compared

to the ideal case of correctly identifying every feature vector in the test set.

$$A = \frac{1}{P} \sum_{i=1}^P \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \quad (2.42)$$

2.5.1.2 Precision

Precision is utilized to calculate how many correct TP classifications were made when compared with the total number of positive classifications, or TP and FP . Precision is given in equation 2.43, where Pr is the Precision.

$$Pr = \frac{1}{P} \sum_{i=1}^P \frac{TP_i}{TP_i + FP_i} \quad (2.43)$$

2.5.1.3 Recall

Recall is a complement to precision. It calculates the number of classifications that were successful compared to those miss classified. The equation for the calculation of recall is given in equation 2.44, where $\mathcal{R}$ is the Recall.

$$\mathcal{R} = \frac{1}{P} \sum_{i=1}^P \frac{TP_i}{TP_i + FN_i} \quad (2.44)$$

2.5.1.4 F_1 Score

Finally, the F_1 score can be calculated using the values obtained with Precision and Recall. It is used to determine how accurate the classifier is by comparing the Precision and Recall. The F_1 score will be considered the best when Precision and Recall are similar, giving a F_1 score that is higher. The F_1 score is calculated using equation 2.45.

$$F_1 = 2 * \frac{P * \mathcal{R}}{P + \mathcal{R}} \quad (2.45)$$

2.5.2 Open Set Metrics

Open set utilizes a CM of size $N \times P$ similar to close set; however, unlike the closed-set N and P are no longer equal. This makes the resulting CM no longer a square matrix. While the shape of an open set CM is different from that of the closed-set, the metrics calculated from the CM are the same, i.e. TP , TN , FP , and FN are still calculated utilizing equations 2.38 - 2.41. Similarly, the statistics utilized in closed-set to evaluate how well the classification algorithm performed are also utilized to evaluate the performance for the known targets.

When evaluating the open set performance, the open set CM is reduced to a 2×2 matrix. Here the classes of the matrix are the known targets and unknown targets. By grouping all of the known and unknown targets, respectively, the performance of the classifier can be evaluated based on how well the two classes are distinguished.

2.5.2.1 Specificity

One evaluation criteria is Specificity. Specificity evaluates a classification algorithm in order to detail how well the negative classifications are accurately identified. The Specificity of the algorithm can be calculated using equation 2.46, where S is the Specificity. When calculating Specificity, the reduced open set CM is used to find the values of TN , FP , TP , and FN . The equations given in 2.38 - 2.41 are still valid for the reduced open set CM. Figure 2.9 shows the reduced CM as well as the locations of TN , FP , TP , and FN .

$$S = \frac{TN}{TN + FP} \quad (2.46)$$

Reduced Open Set Confusion Matrix

	Known	Unknown
Known	TP	FN
Unknown	FP	TN

Figure 2.9: Reduced Open Set Confusion Matrix

2.5.2.2 Youdens Index

With the calculation of the Specificity, the next evaluation criteria is Youdens Index [16]. Youdens index specifies how well the classifier performed in determining if the target was a member of the known training set. Youdens index can range from -1 to 1 , where 1 indicates a perfect classifier and -1 indicates when a classifier cannot classify a target. It is calculated with equation 2.47, where Y is Youdens Index.

$$Y = \frac{TP}{TP + FN} + \frac{TN}{TN + FP} \quad (2.47)$$

3. Methodology

3.1 Data Preparation

The data set being utilized to evaluate the effectiveness of an open set classification algorithm on HRR radar data was collected by the Moving and Stationary Target Acquisition and Recognition (MSTAR) program [24]. This is referred to as the MSTAR data set. The MSTAR data is a collection of HRR data containing two data sets. The first data set was collected at an aspect angle of 15 degrees, while the second was collected at 17 degrees. Both sets contain data collected over 360 degrees for 10 targets. Note that the collection angle interval is not linear as seen in figure 3.1. This figure depicts the non-uniform angle distribution where, the MSTAR index is plotted on the x axis and the collection angle in degrees on the y axis.

Figure 3.1: Non Uniform Angle Distribution

In the data set, 2 of the 10 targets have multiple variants taking the total target count to 22. The data set was collected by taking x-band 1×1 foot resolution complex SAR images of the individual targets and converting the data into range vs angle data. This is accomplished by taking the Inverse Fast Fourier Transform (IFFT) along the cross range of the SAR data. Utilizing this method results in HRR profiles with a target to clutter ratio similar to that of a moving target [24]. Finally, the complex HRR data is normalized. The resulting HRR chip consists of 3 degrees of angle information as seen in figure 3.2. In this figure, it is shown that the MSTAR data set contains multiple chips, with each chip

consisting of a collection of HRR profiles. The amount of range content varies based upon the masking filter applied to the SAR image.

Figure 3.2: MSTAR HRR Chip

In order to minimize the chances of over training the SVM, the MSTAR data is reduced to a collection of windows. The windows are centered around a collection angle and have a predetermined width as seen in figure 3.3. Note there is a different SVM for each window of every training target. The number of windows is determined by the number of available chip angles and the various angles can be used for more than one window. Due to the fact that a window is “slid” based on the chip angle, this window is referred to as a sliding window. In addition to windowing a collection of chips, each chip is also broken into a set number of new range angle collections, where the number of range bins is held constant and the chip is divided based upon the profile angles, as seen in figure 3.4. With large enough amounts of training data there is no need to subdivide the range angle data, as this reduces the amount of information used to create the feature and the accuracy of the training. In the case of the MSTAR data set and the need for large training and validation sets, subdivision is necessary.

Figure 3.3: Sliding Window Implementation

Figure 3.4: Chip Division

3.2 Centroid Alignment

The MSTAR HRR profiles are generated from a mask filtered SAR image. This makes the range information inconsistent across targets as well as between aspect angles, which makes comparisons difficult as seen in the top image of figure 3.5. To overcome this issue, Shaw [2] utilized a centroid alignment method to normalize a comparison of targets in range. The method proposed by Shaw calculated the centroid of the HRR profiles using equation 3.1, where r_i is the i^{th} range location and s_i is the i^{th} scattering response or the value of the range location.

$$r_c = \frac{\sum_{i=1}^N s_i r_i}{\sum_{i=1}^N s_i} \quad (3.1)$$

The centroid of the two HRR profiles being compared are aligned such that the reference and test data are zero padded to be the same size in the range dimension. Additionally,

the centroid of each profile is located at the center of the newly padded data as is seen in the bottom image of figure 3.5. One modification of Shaw's method that was required due to the nature of the SVM is that both centroid aligned profiles need to be of the same length in range. This means that any part of either the reference or test profile that does not overlap is removed from the profile evenly around the center.

Figure 3.5: Centroid Alignment

Since the centroid alignment method was developed for use with a matched filter, the truncation of the data was done on a comparison by comparison basis. While this is effective for a matched filter, an adjustment is required in order to use with an SVM. In order to apply centroid alignment to the various data sets used in the SVM, an SVM reference

profile was created. The SVM reference profile was selected from the first HRR profile in the windowed training data set for the specified target. Once the reference centroid is calculated, all HRR profiles in the training, validation, and test sets associated with the windowed training data are centroid aligned to the reference profile. To align two profiles the calculated centroid of each profile is compared to determine the difference. Based upon this difference the non-reference profile is shifted until both centroids are the same. Finally the two profiles are filtered as described above such that the same number of range bins are always being compared. Once this alignment is completed, a majority of the bias due to a target's location in a frame is removed.

3.3 Training

In order to optimize the selection of the hyperplane for an SVM, it is necessary to provide a set of data that can be used to find the optimal hyperplane. This data is generally referred to as a training set and should provide data for all of the targets that are known in the problem. A subset of the MSTAR data set described in section 3.1 is used to train the SVM. The MSTAR data set was collected at two elevation angles, the first at 15 degrees and the second at 17 degrees. The training data is produced from the 17 degree collection, which is sometimes referred to as the template. Due to the need for a validation set in addition to the training set only a portion of the template collection was used for training. As discussed in section 3.1 the various MSTAR chips are divided into sliding windows, where the window is defined by the center chip angle and contains a specified number of features. The specified number of features are chosen based on their distance from the center angle. For example if a window consists of 21 feature vectors, where 10 are required for training and the center angle is correlated to the 11th feature vector, then the 5th through 15th feature vectors are used for training and the entire selection of 21 features is used for

validation. The implementation of the algorithm can be seen in figure [3.6](#).

Training

Figure 3.6: Training Diagram

*SVM convention requires that only 1's and -1's be used to represent classes. Most SVM libraries do support other class labels for convenience

Figure 3.7: Training Class Creation in OvA configuration

3.4 Validation

The validation data is of utmost importance when attempting to determine if a test feature belongs to a known target. Since the success of open set is closely tied to the ability to properly estimate the CDF of the decision metric, it is important that a sufficiently large amount of data be set aside for the estimation of the CDF. The selection of the validation set utilizes the same method as the training set; however, instead of using only a select number of angles from each window, all data, including the data used in the training set, is selected for the validation set.

3.5 Testing

In order to determine the effectiveness of the feature extraction algorithms as well as the decision metric it is necessary to select a collection of data that is different from the training and validation set for this experiment. The testing data was selected from the 15 degree MSTAR data collection. A testing set is selected for every window described in section 3.3. With the data divided into sliding windows of the same size as the template data set, all features that fall within the window are selected to be in the test set. As a general rule, if there are not enough feature vectors in either the testing or training set, then that center angle is discarded for training and testing. An example of the implementation of the testing method is given in figure 3.8.

Testing

Figure 3.8: Testing Diagram

3.6 Kernel Selection

Currently, one of the biggest challenges with the nonlinear SVM is how to best choose which kernel function to use. Although there have been some specialized methods developed to determine the optimal kernel function as well as kernel parameters, their reliability is low and generally can not be extended beyond the scope of the data used to test them. It is for this reason that the accepted method for optimizing the selection of the Kernel function and its parameters is to use an iterative approach with cross validation. This means that the training set or validation set when available is used to determine which kernel provides the greatest classification rate. Cross validation is performed using a set of data, either the training or validation set, with known classes such that the SVM can be optimized without biasing the outcome by finding the optimal parameters for the test set.

The accepted optimization method is to run a search over a sampled set of all possible configuration. In the following experiments the data was optimized using a validation set to find the optimal kernel parameters for the Radial Bias Function (RBF), or Gaussian Kernel. An Example of the Kernel Optimization is given in figure [3.9](#)

3.7 Threshold Selection

One of the requirements for an open set classification algorithm is the idea of a rejection. Ideally, an open set classifier would be able to algorithmically determine when to label a target as unknown without any input from a user. However, this is not generally how classification algorithms work with a threshold being chosen which determines when a target is labeled as known or unknown. Scherrek in [\[16\]](#) created an iterative method for finding the ideal threshold based on the affect a selection of the threshold has on the open set risk, which was derived by Scheirer [\[28\]](#). While this threshold selection algorithm worked well for the data sets used in [\[16\]](#), it was found to be unsuccessful on HRR data. Instead,

an estimate of the ideal threshold was made utilizing a user defined miss rate and a fit to the cross validated SVM scores. To make the estimate of the ideal threshold, a histogram is formed from the scores returned by cross validating the optimized SVM. The histogram count is then normalized to form a CDF. The cumulative sum of the CDF is taken and the bin at which the cumulative sum of the CDF is closest to the miss rate is chosen as the threshold.

SVM Optimization using Training and Validation Data

Figure 3.9: SVM Kernel Optimization

4. Algorithm Evaluation

The evaluation of the POS-SVM as a multi-class OSR algorithm for HRR data is below. Utilizing the MSTAR data set [3.1](#), several test scenarios were created to evaluate utilizing the ET and MT feature recognition algorithms on closed and open multi-class configurations. The proposed feature extraction method allows for the optimization of several variables in order to take full advantage of the POS-SVM for OSR and the non-linear SVM for closed-set recognition. As defined in section [3.1](#), there are two collection angles in the MSTAR data set. The first collection taken at 15 degrees is utilized in the training and optimization of the SVM as well as for the validation set, which is used in the probability estimate of the classification of a feature vector. Due to the nature of the SVM, an iterative optimization is required over the tunable parameters of the SVM. The parameters which can be optimized in the SVM are the Kernel as well as each kernels associated coefficients. The POS-SVM utilizes the RBF or Gaussian kernel to train the data, which has a sigma coefficient utilized to indicate the amount of variance.

4.1 Closed Set

Utilizing the optimization method for the training of the SVM, it is possible to determine how the various training variables affect the performance of the SVM. The training variables that are applicable for closed set or forced decision classification are: the number of profiles to use for testing, the chip division rate, the number of features to keep after cen-

troid alignment, the size of the training window in degrees, and the number of profiles to require within each training window. The optimization of these 5 parameters will provide a better understanding as to the effectiveness of the SVM as an HRR classifier as well as the effectiveness of the ET and MT feature extraction algorithms. The ideal values found with these methods will provide general guidance for the optimal setting for a closed-set ATR on HRR data.

The number of feature vectors utilized for training is very important when it comes to the SVM. Without the proper number of training points, the SVM hyperplane will be a poor estimate of the actual division of classes in the feature space. However, if too many points of data are included for training, there will be little separation of the classes. Therefore it is important that only unique data is utilized to train the SVM. The nature of the MT as well as the ET feature extraction methods is to reduce a collection of HRR profiles or a chip into a single feature vector. While it is optimal to utilize a large number of HRR profiles with the ET method, the requirement for a large amount of training data requires that the number of feature vectors that is gathered from a chip be increased. To facilitate this need for more training data, a single HRR radar chip is divided into sub-chips with sub-chip angles associated with the divisions relative location within the chip from which it was derived. This method effectively increases the number of feature vectors that can be created from the data set. In figure 4.1, it can be seen that the need to divide the chip into a large number of sub-chips is not required for ET. The MT results seen in figure 4.2 show that the number of chip divisions as well as the number of training features does affect the accuracy of SVM trained with the MT features. While the change in accuracy is not drastic, it is important to note that the fewer the subdivisions the higher the accuracy. This figure goes on to show that the number of training features has a direct affect on the accuracy of the SVM.

Figure 4.1: Chip Division and Number of Training Features vs Accuracy for SVM - Eigen-Template

Figure 4.2: Chip Division and Number of Training Features vs Accuracy for SVM - Mean Template

It is important to remember when selecting the number of chip divisions that when fewer divisions are created the number of angles that have enough data to be used goes down. This is represented in figure 4.3, where the number of train features and chip divisions is plotted against the number of angles for which data is available. Due to the lack of need for chip divisions in both the ET and MT results, it is important that the number of angles be monitored. It is also important to remember that as the number of training profiles goes up, the number of angles will again go down.

Figure 4.3: Chip Division and Number of Training Features Accuracy vs Number of Utilized Angles

When training the SVM, an important step is the centroid alignment of the feature vectors. Due to the nature of HRR data, the number of features collected from a target can vary, which makes classification biased to those feature vectors of similar length. To remove this bias, the number of features which are used in training is reduced, where features are removed evenly about the center of the vector until the vector has reached the determined length. Figures 4.4 and 4.5 show how the number of features affects the accuracy of the classifier for both the ET and MT feature extraction methods respectively. The ET method is relatively unaffected by a change in the number of features and the MT method is negatively affected by an increase in the number of features used.

Figure 4.4: Accuracy vs Number of Features for SVM - Eigen Template

Figure 4.5: Accuracy vs Number of Features for SVM - Mean Template

Having optimized chip division, the number of training profiles, and the number of features, the next optimization is the window size. In figures 4.6 and 4.7, we see that the size of the window has an inverse effect on the accuracy of the SVM for both the ET and MT feature extraction algorithms. The decrease in accuracy in the closed-set is due to the nature of how the HRR profiles and therefore, feature vectors are formed. In section 2.1, it is seen that each HRR profile is formed based upon the chip angle as well as the combination of the signals reflected from the surface of the target. This means that the HRR profiles can be thought of as independent of each other in angle. In order to reduce

the chance of calculating a false positive, it is important that the HRR profiles be compared with a small collection of HRR profiles taken around the same chip angle. Again, the selection of the window size must also take into consideration the effect of shrinking the number of angles for which enough data exists.

Figure 4.6: Accuracy vs Window Size for the SVM-Eigen Template

Figure 4.7: Accuracy vs Window Size for the SVM - Mean Template

The final optimization to consider is the number of test profiles to be used for classification. In equation 2.24, we see that the classification of a target requires the comparison of a test vector with the trained SVM. The importance of the proper selection of the number of test profiles is due to the way profiles are gathered. Since each chip represents a collection of data from a target, it is ideal that the number of chips or profiles within each chip required to make a classification be small in order to reduce the amount of information needed to classify a target as well as to reduce the computational time. In figure 4.8 and 4.9, we see how the accuracy of the SVM is affected by the number of test profiles for the ET and MT feature extraction methods. For the ET method, we can see that the number of test profiles used has no real affect on the accuracy of the SVM. However, the number of profiles used for the MT method does have an affect on the accuracy, with a higher number of profiles being desired.

Figure 4.8: Accuracy vs Number of Test Profiles SVM-Eigen Template

Figure 4.9: Accuracy vs Number of Test Profiles SVM - Mean Template

The optimal parameters for the training and testing of the MSTAR data set are described in table 4.1 for the ET SVM and in table 4.1 for the MT SVM.

Closed Set Eigen Template Parameters

<i>Parameters</i>	<i>Value</i>
Number Of Divisions	5
Number of Training Profiles	18
Number of Features per Profile	63
Number of Test Profiles per Test Angle	7

Table 4.1: Closed Set Tuning Parameters SVM-Eigen Template

Closed Set Mean Template Parameters

<i>Parameters</i>	<i>Value</i>
Number Of Divisions	5
Number of Training Profiles	20
Number of Features per Profile	60
Number of Test Profiles per Test Angle	30

Table 4.2: Closed Set Tuning Parameters SVM - Mean Template

With the SVM trained in this way, a statistical analysis can be performed on the results for the closed set. The resulting CM for the ET and MT feature extraction methods is given in table 4.1 and 4.1 respectively.

Closed Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	brdm2_e71
bmp2_c21	.9120	.0080	.08	0
t72_132	0	.9943	0	.0057
2s1_b01	.1242	.0030	.8697	.0030
brdm2_e71	.0047	0	0	.9953

Table 4.3: Closed Set Confusion Matrix for SVM-Eigen Template Features

Closed Set Mean Template Confusion Matrix

	bmp2_c 21	t72_132	2s1_b0 1	brdm2_e71
bmp2_c21	.8	.048	.1307	.0213
t72_132	.0114	.8917	.0513	.0456
2s1_b01	.1061	.1106	.7242	.0591
brdm2_e71	.0218	.0591	.0653	.8538

Table 4.4: Closed Set Confusion Matrix for SVM - Mean Template Features

With the optimal parameters found for the closed set SVM on HRR data being determined and the CM calculated, it is possible to evaluate the performance of the SVM with the ET and MT feature extraction techniques. The analysis of the SVM is performed by calculating the accuracy of the SVM as well as the F1 Score of the classifier. In figure 4.10, we see that results for both the MT and ET feature extraction methods. In table 4.5, we see the corresponding scores.

Figure 4.10: Closed set Accuracy and F1 Score

Closed Set Results

	Eigen	Mean
Accuracy	93.886	85.550
F1	.9364	.8501

Table 4.5: Closed Set Accuracy and F1 Score

Based on the results above, it is evident that the ET SVM is superior to the MT SVM. However, the choice of the SVM as an ideal closed set classifier is yet to be determined. The ET as a feature extraction algorithm is well developed [2][7][9]. However, the application of the ET method to an SVM has not been performed prior to this thesis. In order to determine the viability of the SVM as a HRR classifier, it is important that the results obtained above be compared with earlier work. Shaw in [2] utilized the ET method in combination with a matched filter as a classifier for HRR data. The Eigen Template Matched Filter (ETMF) developed in [2], was found to have a *PCC* or Accuracy of 81.5%, where the CM for the ETMF is seen in figure 4.11 for the same targets as tested above in a similar scenario where only a single test vector is used for classification. The ET SVM has been shown to have a *PCC* of 93.8% which is superior to the ETMF in [2]. Based on these results, the ET feature extraction algorithm in combination with an SVM is proposed as a good closed set classifier for HRR data.

TABLE IV
Confusion Matrix for ETMF with Single Profile Testing
(PCC = 81.5%)

Target	BMP2	T72	2S1	BRDM2
BMP2	0.7727	0.0876	0.0682	0.0715
T72	0.0780	0.8330	0.0435	0.0455
2S1	0.0649	0.0334	0.8233	0.0784
BRDM2	0.0557	0.0474	0.0700	0.8269

Figure 4.11: SVM - Eigen Template Matched Filter Confusion Matrix [2]

In addition to the results above, further tests were run to show the algorithms effectiveness when trained and tested against more targets. These results are seen in tables 4.6 - 4.8. It is observed that adding more targets does decrease the *PCC*; however, the algorithm is still able to distinguish the targets better than the ETMF results seen in table 4.11.

Closed Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	brdm2_e71	zil131_e12
bmp2_c21	.8933	.0080	.0647	0	.0347
t72_132	.0028	.9915	0	.0057	0
2s1_b01	.1091	.0015	.7576	.0015	.1303
brdm2_e71	.0078	0	0	.9922	0
zil131_e12	.0509	.0031	.1343	.0031	.8086

Table 4.6: Closed Set SVM - Eigen Template Confusion Matrix for 5 Targets

Closed Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	brdm2_e71	zil131_e12	zsu234_d08
bmp2_c21	.8320	.0080	.0587	.0053	.0427	.0533
t72_132	0	.9943	0	.0057	0	0
2s1_b01	.0970	.0015	.7485	0	.1318	.0212
brdm2_e71	0	0	0	.9767	.0062	.0171
zil131_e12	.0417	.0015	.1281	.0015	.7932	.0340
zsu234_d08	.0669	.0031	.0156	.0358	.0435	.8351

Table 4.7: Closed Set SVM - Eigen Template Confusion Matrix for 6 Targets

Closed Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	brdm2_e71	zil131_e12	zsu234_d08	d7_92v13015
bmp2_c21	.8195	0	.0686	.0036	.0578	.0505	0
t72_132	0	.9920	0	.0080	0	0	0
2s1_b01	.1102	.0020	.7315	0	.1303	.0261	0
brdm2_e71	0	0	0	.9430	.0183	0	.0387
zil131_e12	.0631	.0041	.0143	.0387	.8289	.0448	.0061
zsu234_d08	.0463	.0020	.1429	.0020	.0423	.7646	0
d7_92v13015	0	0	0	.05	.0167	0	.9333

Table 4.8: Closed Set SVM - Eigen Template Confusion Matrix for 7 Targets

4.2 Open Set

The same optimization method described in section 4.1 will be applied for optimization of POS-SVM with one exception. When determining how well an OSR algorithm performed, it is important to identify the effectiveness of the algorithms ability to distinguish known targets from unknown targets, as described in section 2.4. In addition to the training variables described in closed set classification, one additional variable must be optimized. A classification threshold must be chosen such that it can be used to reject a target that is not in the training set. As described in section 3.7, the SVM score can be used to compute the probability that a target belongs to the selected class. The ideal values found with these methods will provide general guidance for the optimal setting for open set ATR on HRR data.

The optimization of the number of training feature vectors and HRR chip divisions affects the POS-SVM in a much different way than it affects the SVM. In figure 4.12 and 4.13, it is seen that more divisions are required in order for the POS-SVM to properly classify those targets which are known to the POS-SVM. However, the affect on the unknown target accuracy for an increase in the number of divisions is shown to be negative. Based on these results, a higher number of divisions is desired. The need for more feature vectors by the POS-SVM is a result of the requirement that the estimation of the probability of a SVM score belonging to a class requires a large validation data set. The affect that the number of training features and HRR chip division has on the POS-SVM Mean Template is seen in figures 4.14 and 4.15. Based on these figures, we can see that neither the number of training feature nor the number of chip divisions affects the known or unknown target accuracy.

Figure 4.12: Chip Division and Number of Training Features vs Known Target Accuracy for POS-SVM Eigen Template

Figure 4.13: Chip Division and Number of Training Features vs Unknown Target Accuracy for POS-SVM Eigen Template

Figure 4.14: Chip Division and Number of Training Features vs Known Target Accuracy for POS-SVM Mean Template

Figure 4.15: Chip Division and Number of Training Features vs Unknown Target Accuracy for POS-SVM Mean Template

In figures 4.16 - 4.19, it can be seen that for the POS-SVM - Eigen Template the number of features has little affect on the accuracy once greater than 65 range bins. However, for the POS-SVM - Mean Template it is shown that having a large number of range bins per feature is a disadvantage.

Figure 4.16: Accuracy vs Number of Features POS-SVM Eigen Template

Figure 4.17: Unknown Target Accuracy vs Number of Features when trained with POS-SVM Eigen Template Data

Figure 4.18: Accuracy vs Number of Features POS-SVM Mean Template

Figure 4.19: Unknown Target Accuracy vs Number of Features when trained with Mean Template Data

The window size utilized for training the SVM is shown to have a different affect on the accuracy for the POS-SVM - Eigen Template than that of the SVM, as shown in figure 4.20. A desire for a higher known target accuracy would drive the selection of the window size to balance the unknown classification accuracy. The accuracy vs window size for the POS-SVM - Mean Template is seen in figure 4.21. In this figure, the ideal window size is small compared to the POS-SVM Eigen Template due to a sharp decline in known target accuracy.

Figure 4.20: Accuracy vs Window Size POS-SVM Eigen Template

Figure 4.21: Accuracy vs Window Size for the POS-SVM Mean Template

Figure 4.22: Accuracy vs Number of Test Profiles POS-SVM - Eigen Template

The final optimization performed in both the closed and open set is the choice of the number of test profiles used during classification seen in figures 4.22 and 4.23. The effect that the number of test profiles has on the data is generally minimal for the SVM - Eigen Template as the change in accuracy is minor. However, due to the desire to improve the known target accuracy, a higher number of test profiles should be chosen for the ET method. The POS-SVM - Mean Template does not show any real information about how changing the number of test profiles affects the accuracy of the algorithm since the known target accuracy is quickly less than 50%.

Figure 4.23: Accuracy vs Number of Test Profiles POS-SVM - Mean Template

The parameter that has the largest affect on the accuracy of the known and unknown classification is that of the threshold. Since the POS-SVM relies on a properly chosen threshold value to balance the classification of known and unknown targets, its choice is important to determine what kinds of errors are acceptable in the classification algorithm. While the selection of the proper T values is chosen by an algorithm, the probability of false alarm is set by the user to determine what amount of error is acceptable. To determine the ideal threshold, a similar optimization is performed. In figure 4.24 and 4.25, we can see where the unknown accuracy and known accuracy cross will give a good balance for the known target vs unknown target classification. Depending on the desired false alarm rate of the system, we can determine what the ideal threshold would be.

Figure 4.24: Known and Unknown Accuracy vs Threshold for POS-SVM - Eigen Template

Figure 4.25: Known and Unknown Accuracy vs Threshold for POS-SVM - Mean Template

The Receiver Operating characteristics (ROC) curve shown in figures 4.26 and 4.27, for the ET and MT methods, shows that the POS-SVM Eigen Template does a decent job of identifying when a target is unknown. The POS-SVM - Mean Template does not perform as well as the ET method. In figure 4.28, the averaged ROC curve is shown for both the ET and MT methods. This was found by averaging the target results for each threshold. One interesting result is that a perfect probability, of classification is never reached due to the fact that the threshold is based on a probability which only has a range from 0 to 1.

Figure 4.26: ROC Curve for POS-SVM - Eigen Template by Target

Figure 4.27: ROC Curve for POS-SVM - Mean Template by Target

Figure 4.28: Averaged ROC Curve for POS-SVM ET and MT

Based upon the optimizations shown above, the parameters given in table 4.9 are used to classify HRR data with an open set ATR algorithm for the POS-SVM - Eigen Template. Similarly, the parameters given in table 4.10 are used for the POS-SVM - Mean Template.

Open Set Eigen Template Parameters

<i>Parameters</i>	<i>Value</i>
Number Of Divisions	10
Number of Training Profiles	20
Number of Features per Profile	68
Number of Test Profiles per Test Angle	30
Miss Rate	10%

Table 4.9: Open Set POS-SVM - Eigen Template Tuning Parameters

Open Set Mean Template Parameters

<i>Parameters</i>	<i>Value</i>
Number Of Divisions	5
Number of Training Profiles	18
Number of Features per Profile	60
Number of Test Profiles per Test Angle	30
Miss Rate	10%

Table 4.10: Open Set POS-SVM - Mean Template Tuning Parameters

Open Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	Unknown
bmp2_c21	.7596	.0082	.1181	.1140
t72_132	.0098	.95	.0049	.0352
2s1_b01	.0871	.0036	.8172	.0921
Unknown	.1121	.0277	.0345	.8257

Table 4.11: Open Set Confusion Matrix POS-SVM - Eigen Template

Open Set Mean Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	Unknown
bmp2_c21	.6411	.0117	.0941	.2531
t72_132	.0067	.7366	.0301	.2266
2s1_b01	.0624	.0086	.7016	.2274
Unknown	.0195	.0572	.2521	.6711

Table 4.12: Open Set Confusion Matrix POS-SVM - Mean Template

With the POS-SVM trained in this way, a statistical analysis can be performed on the results for the open set. The resulting CM for the POS-SVM - Eigen Template as well as the POS-SVM - Mean Template is given in tables 4.11 and 4.12 respectively. Additionally, the reduced open set CM is given in figures 4.13 and 4.14.

Reduced Open Set Eigen Template Confusion Matrix

	Known	Unknown
Known	.9181	.0819
Unknown	.2668	.7332

Table 4.13: Reduced Open Set Confusion Matrix POS-SVM - Eigen Template

Reduced Open Set Mean Template Confusion Matrix

	Known	Unknown
Known	.748	.2519
Unknown	.3288	.6711

Table 4.14: Reduced Open Set Confusion Matrix POS-SVM - Mean Template

With the optimal parameters found for the open set SVM on HRR data being determined and the CM calculated, it is possible to evaluate the performance of the POS-SVM with the ET and MT feature extraction techniques. The analysis of the POS-SVM is performed by calculating the accuracy of the classifier as well as the F1 Score of the classifier with the results shown in figure 4.29. Additionally, the accuracy and Youden Index for the reduced open set CM are calculated to determine how well the classifier distinguished known from unknown targets, as seen in figure 4.30. In figure 4.10, we see the results for both the MT and ET feature extraction methods. In table 4.5, we see the corresponding scores. One interesting result seen in figure 4.30 is that the Youden Index for the MT is higher than the ET indicating that the MT scored better. This result is due to the difference between the known and unknown PCC being lower for the MT when compared to the ET despite the values being lower than the ET.

Figure 4.29: Open Set Accuracy and F1 Score for Known Targets

Figure 4.30: Open Set Classification Accuracy

Open Set Results

	Eigen	Mean
Accuracy Known	.8342	.6422
F1	.8675	.7563
Accuracy	.8548	.7485
Youden Index	.6018	.8509

Table 4.15: Open Set Statistics

In addition to the results above, further tests were run to show the POS-SVM Eigen Template's effectiveness when trained for more targets as well as to determine the performance when more than one confuser was present. These results are seen in tables 4.16 - 4.19. It is observed that adding more targets does decrease the probability of correctly classifying a target, the algorithm is still able to distinguish known targets from unknown targets.

Open Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	brdm2_e71	Unknown
bmp2_c21	.6555	.0168	.0771	.0101	.2406
t72_132	.0025	.7441	.0119	.00008	.2407
2s1_b01	.0637	.0034	.6954	.0086	.2289
brdm2_e71	.0076	.0048	.0129	.7917	.1830
Unknown	.0758	.0143	.1082	.0896	.7121

Table 4.16: Open Set Confusion Matrix 4 Known and 1 Unknown POS-SVM - Eigen Template

Open Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	brdm2_e71	zil131_e12	Unknown
bmp2_c21	.6764	.0193	.0721	.0042	.0645	.1635
t72_132	.0034	.7847	.0102	0	.0085	.1932
2s1_b01	.0685	.0077	.6753	.0053	.0627	.1806
brdm2_e71	.0038	.0095	.0052	.7893	.0577	.1344
zil131_e12	.0281	.0057	.0191	.0324	.7684	.1463
Unknown	.0505	.0157	.3308	.0234	.1392	.4404

Table 4.17: Open Set Confusion Matrix 5 Known and 1 Unknown POS-SVM - Eigen Template

Open Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	brdm2_e71	Unknown
bmp2_c21	.6555	.0168	.0771	.0101	.2406
t72_132	.0025	.7441	.0119	.00008	.2407
2s1_b01	.0637	.0034	.6954	.0086	.2289
brdm2_e71	.0076	.0048	.0129	.7917	.1830
Unknown	.0758	.0143	.1082	.0896	.7121
Unknown	.0524	.01	.3599	.0381	.5396

Table 4.18: Open Set Confusion Matrix 4 Known and 2 Unknown POS-SVM - Eigen Template

Open Set Eigen Template Confusion Matrix

	bmp2_c21	t72_132	2s1_b01	Unknown
bmp2_c21	.6411	.0117	.0941	.2531
t72_132	.0067	.7366	.0301	.2266
2s1_b01	.0624	.0086	.7016	.2274
Unknown	.0195	.0572	.2521	.6711
Unknown	.0801	.0319	.2312	.6568
Unknown	.0534	.0214	.4156	.5095

Table 4.19: Open Set Confusion Matrix 3 Known and 3 Unknown POS-SVM - Eigen Template

5. Conclusion

By taking advantage of feature extraction algorithms previously developed in the closed-set and the capabilities of the SVM as a classifier, it is possible to perform ATR on HRR targets with a higher *PCC* than previous algorithms. The results detailed in chapter 4 show that the Support Vector Machine with Eigen Template performs well as a closed set classifier of HRR data when trained with the RBF kernel, attaining an accuracy of 93.8% . In contrast, the Support Vector Machine with Mean Template did not perform as well as the SVM-Eigen Template, which only attains a *PCC* of 85.5%.

In addition to the Eigen Template method being ideal for the SVM, it was also found to be ideal for the POS-SVM. By utilizing the POS-SVM - Eigen Template, it is possible to successfully perform open set recognition on HRR data. The POS-SVM - Eigen Template was able to achieve a *PCC* of 83.8% for known targets and a *PCC* of 82.5% for unknown targets. Ideally, additional testing would be performed to substantiate these results with more data sets. However, these results indicate that the POS-SVM - Eigen Template is a good choice as a OSR classifier for HRR data.

Bibliography

- [1] J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” *Advances In Large Margin Classifiers*, pp. 61–74, 1999.
- [2] A. K. Shaw, A. S. Paul, and R. Williams, “Eigen-template-based HRR-ATR with multi-look and time-recursion,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, pp. 2369–2385, OCTOBER 2013.
- [3] R. L. Williams, D. C. Gross, J. J. Westerkamp, A. P. Palomino, and D. Wardell, “1D HRR data analysis and ATR assessment,” *Proc. SPIE, Algorithms for Synthetic Aperture Radar Imagery*, vol. 3370, pp. 588–599, 1998.
- [4] B. Pei and Z. Bao, “Radar target recognition based on peak location of HRR profile and HMMs classifiers,” in *RADAR 2002*, pp. 414–418, Oct 2002.
- [5] Q. Zhao and J. C. Principe, “Support vector machines for SAR automatic target recognition,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, pp. 643–654, Apr 2001.
- [6] B. Chen, H. Liu, and Z. Bao, “PCA and kernel PCA for radar high range resolution profiles recognition,” in *IEEE International Radar Conference, 2005.*, pp. 528–533, May 2005.

- [7] A. K. Shaw and V. Bhatnagar, "Automatic target recognition using eigen templates," *Proc. SPIE Algorithms for Synthetic Aperture Radar Imagery*, vol. 3370, pp. 448–459, 1998.
- [8] V. Bhatnagar, A. K. Shaw, and R. W. Williams, "Improved automatic target recognition using singular value decomposition," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 5, pp. 2717–2720 vol.5, May 1998.
- [9] A. K. Shaw, R. Vashist, and R. L. Williams, "HRR ATR using eigen templates with noisy observations in unknown target scenario," *Proc. SPIE Algorithms for Synthetic Aperture Radar Imagery*, vol. 4053, pp. 467–478, 2000.
- [10] A. S. Paul, A. K. Shaw, K. Das, and A. K. Mitra, "Improved HRR-ATR using hybridization of HMM and eigen-template-matched filtering," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, vol. 2, pp. II–397–400 vol.2, April 2003.
- [11] A. S. Paul and A. K. Shaw, "Robust HRR radar target identification by hybridization of HMM and eigen-template-based matched filtering," *Proc. SPIE Automatic Target Recognition XIII*, vol. 5094, pp. 278–289, 2003.
- [12] J. Ma, *Feature Study for High-Range-Resolution Based Automatic Target Recognition: Analysis and Extraction*. PhD thesis, The Ohio State University, 2001.
- [13] J. Ma, H. Li, and S. C. Ahalt, "Using support vector machines as hrr signature classifiers," *Proc. SPIE Automatic Target Recognition XI*, vol. 4379, pp. 209–215, 2001.
- [14] H. Li, J.-L. Sancho Gomez, and S. C. Ahalt, "Kernel-based discriminant feature extraction using a representative dataset," *Proc. SPIE Signal Processing, Sensor Fusion, and Target Recognition XI*, vol. 4729, pp. 352–363, 2002.

- [15] J. Gudnason, “HRR automatic target recognition from super-resolution scattering center features,” *IEEE Transactions on Aerospace and electronic systems*, vol. 45, no. 4, pp. 1512–1524, 2009.
- [16] M. D. Scherreik, “A Probabilistic Technique For Open Set Recognition Using Support Vector Machines,” MS Thesis, Wright State University, Dayton, OH, 2014.
- [17] M. D. Scherreik and B. D. Rigling, “Open set recognition for automatic target classification with rejection,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, pp. 632–642, April 2016.
- [18] M. Scherreik and B. Rigling, “Ladar ATR via probabilistic open set techniques,” *SPIE Defense+ Security. International Society for Optics and Photonics*, vol. 9090, pp. 90900A–90900A–9, 2014.
- [19] M. Scherreik and B. Rigling, “An evaluation of open set recognition for FLIR images,” *SPIE Defense+ Security. International Society for Optics and Photonics*, vol. 9476, pp. 94760N–94760N–10, 2015.
- [20] M. Scherreik and B. Rigling, “Multi-class open set recognition for SAR imagery,” *SPIE Defense+ Security. International Society for Optics and Photonics*, vol. 9844, pp. 98440M–98440M–9, 2016.
- [21] M. I. Skolnik, “Radar.” <https://www.britannica.com/technology/radar>, 2015. Accessed: 2016-09-11.
- [22] D. Nelson, *High Range Resolution Radar Target Classification: A Rough Set Approach*. PhD thesis, The Ohio State University, 2001.
- [23] M. A. Temple, K. L. Sitler, R. A. Raines, and J. A. Hughes, “High range resolution (HRR) improvement using synthetic HRR processing and stepped-frequency

- polyphase coding,” *IEEE Proceedings - Radar, Sonar and Navigation*, vol. 151, pp. 41–47, Feb 2004.
- [24] A. Palomino, “README - MSTAR.” README.
- [25] V. Vapnik, *Estimation of Dependences Based on Empirical Data*. New York, NY: Springer-Verlag New York, 1982.
- [26] A. Moore, “Support vector machines.” University Lecture, <http://www.autonlab.org/tutorials/svm15.pdf>, 2001.
- [27] M. Al-Haj and H. Hel-Or, “Support vector machines.” University Lecture, http://www.cs.haifa.ac.il/hagit/courses/seminars/visionTopics/Presentations/SVM_Lecture.pptx.
- [28] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, “Toward open set recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 1757–1772, July 2013.
- [29] S. Theodoridis and K. Konstantinos, *Pattern Recognition*. Burlington, MA: Academic Press, 4 ed., 2009.